

Novática, revista fundada en 1975 y decana de la prensa informática española, es el órgano oficial de expresión y formación continua de **ATI** (Asociación de Técnicos de Informática). **Novática** edita también **UPGRADE**, revista digital de **CEPIS** (Council of European Professional Informatics Societies), en lengua inglesa, y es miembro fundador de **UPENET** (**UPGRADE European Network**)

<<http://www.ati.es/novatica/>>
<<http://www.upgrade-cepis.org/>>

ATI es miembro fundador de **CEPIS** (Council of European Professional Informatics Societies) y es representante de España en **IFIP** (International Federation for Information Processing); tiene un acuerdo de colaboración con **ACM** (Association for Computing Machinery), así como acuerdos de vinculación o colaboración con **AdaSpain**, **AIZ** y **ASTIC**.

Consejo Editorial

Antoni Carbonell Noguera, Juan Manuel Cueva Lovelle, Juan Antonio Esteban Iriarte Francisco, Juan Crespo, Celestino Martín Alonso, Josep Molas i Bertrán, Olga Pallás Codina, Fernando Píera Gómez (Presidente del Consejo), Ramón Puigjaner Trepal, Miquel Sàrries Griño, Asunción Yturbe Herranz

Coordinación Editorial

Rafael Fernández Calvo <rfcalvo@ati.es>

Composición y autedición

Jorge Lázcar Gil de Ramales

Traducciones
Grupo de Lengua e Informática de ATI <<http://www.ati.es/gl/lengua-informatica/>>

Administración

Tomás Brunete, María José Fernández, Enric Camarero, Felicidad López

Secciones Técnicas: Coordinadores

Administración Pública electrónica
Gumersindo García Arribas, Francisco López Crespo (MAP)

<gumersindo.garcia@map.es>, <flc@ati.es>

Arquitecturas

Jordi Tubella Murgadas (DAC-UPC) <jordit@ac.upc.es>

Victor Viñals Yufiera (Univ. de Zaragoza) <victor@unizar.es>

Auditoría ética

Marina Tourinho Trullitro, Manuel Palao García-Suelto (ASIA)

(Escuela Superior de Informática, UCLM) <manuel@palao.com>

Bases de datos

Coral Calero Muñoz, Mario G. Plattini Velthuis

(Escuela Superior de Informática, UCLM) <Coral.Calero@uclm.es>, <mplattini@inf-cr.uclm.es>

Derecho e tecnologías

Isabel Hernández Colados (Fac. Derecho de Donostia, UPV) <ihernando@legaltek.net>

Elena Davara Fernández Marcos (Davara & Davara) <edavara@davara.com>

Enseñanza Universitaria de la Informática

Joaquín Ezepeleta Mateo (CPS-UZAR) <ezepeleta@posta.unizar.es>

Cristóbal Pareja Flores (DSP-UCM) <cpareja@sip.ucm.es>

Gestión del Conocimiento

Juan Baiget Solé (Cap Gemini Ernst & Young) <juan.baiget@ati.es>

Informática y Filosofía

José Corco Jorinúa (UC) <jcorco@unica.edu>

Esperanza Marcos Martínez (ESCEIT-URJC) <esceita@escet.urjc.es>

Informática Gráfica

Miguel Chover Solís (Universitat Jaume I de Castellón) <chover@lsi.uji.es>

Roberto Vivó Hernández (Eurographics, sección española) <rvivo@dsic.upv.es>

Ingeniería del Software

Javier Dolado Cosín (DLSI-UPV) <dolado@si.ehu.es>

Luis Fernández Sanz (PDSI-EJ-UEM) <lfern@dpriis.esi.uem.es>

Inteligencia Artificial

Federico Barber Sanchis, Vicente Botti Navarro (DSIC-UPV)

<fvotti_barber@fse.upv.es>

Interacción Persona-Computador

Julio Abascal González (FI-UPV) <julio@si.ehu.es>

Jesús Lorés Vidal (Univ. de Lleida) <jesus@eup.udl.es>

Internet

Alonso Álvarez García (TID) <alonso@ati.es>

Llucen Pagés Casas (Indra) <pages@ati.es>

Lengua e Informática

M. del Carmen Ugarte García (IBM) <cuarte@ati.es>

Lenguajes Informáticos

Andrés Martín López (Univ. Carlos III) <amarin@it.uc3m.es>

J. Angel Velázquez Hurtado (ESCEIT-URJC) <a.velazquez@escet.urjc.es>

Librerías e Informática

Alonso Escolano (FIR-Univ. de La Laguna) <aescolano@ull.es>

Xavier Gómez Guinovart (Univ. de Vigo) <xgg@uvigo.es>

Manuel Palomar (Univ. de Alicante) <mpalomar@disi.ua.es>

Mundo estudiantil

Adolfo Vázquez Rodríguez (Rama de Estudiantes del IEEE-UCM)

<a.vazquez@ieee.org>

Profesión Informática

Rafael Fernández Calvo (ATI) <rfcalvo@ati.es>

Miquel Sàrries Griño (Ayto. de Barcelona) <msarries@ati.es>

Redes y servicios telemáticos

Luis Gujardo Colombia (DCOM-UPV) <lgujard@doom.upv.es>

José Solís Pareta (DAC-UPC) <pareta@ac.upc.es>

Seguridad

Javier Arellito Bertolin (Univ. de Deusto) <jarellito@eside.deusto.es>

Javier López Muñoz (ETSI Informática-UMA) <jlm@lcc.uma.es>

Sistemas de Tiempo Real

Alejandro Alonso Muñoz, Juan Antonio de la Puente Añaró (DIT-UPM)

<aalonso.igunte@di.upm.es>

Software Libre

Jesus M. González Barahona, Pedro de las Heras Quirós

(GSYC-URJC) <frob.gheras@gsyc.esct.urjc.es>

Teconología de Objetos

Jesus Garcia Molina (DIS-UM) <jmolina@correo.um.es>

Gustavo Rossi (LIFIA-UNLP, Argentina) <gustavo@sol.info.unlp.edu.ar>

Teconologías para la Educación

Juan Manuel Dodero Barro (UC3M) <dodero@inf.uc3m.es>

Teconologías y Empresa

Pablo Hernández Medrano (Bluemart) <pablohm@bluemart.biz>

TIC para la Gestión
Valentín Masero Vargas (DI-UNEX) <vmasero@unex.es>

TIC y Turismo

Andrés Aguayo Maldonado, Antonio Guevara Plaza (Univ. de Málaga)

<aguayo.guevara@cc.uma.es>

Coordinación Editorial, Redacción Central y Redacción ATI Madrid

Padilla 66, 3º, dcha., 28006 Madrid

Tfn. 914029391; fax 913093685 <novatica@ati.es>

Composición, Edición y Redacción ATI Valencia

Av. del Reino de Valencia 23, 46005 Valencia

Tfn./fax 963300392 <secretgati@ati.es>

Administración y Redacción ATI Cataluña

Ciudad de Granada 131, 08018 Barcelona

Tfn. 934125235; fax 934127113 <secretgen@ati.es>

Redacción ATI Andalucía

Isaac Newton, s/n, Ed. Sadiel,

Isla Cartuja 41092 Sevilla, Tfn./fax 954460779 <secretand@ati.es>

Redacción ATI Aragón

Lagasca 9, 3-B, 50006 Zaragoza,

Tfn./fax 976235181 <secretara@ati.es>

Redacción ATI Asturias-Cantabria

<gp-astucant@ati.es>

Redacción ATI Castilla-La Mancha

<gp-clmancha@ati.es>

Redacción ATI Galicia

Recinto Ferial s/n, 36540 Silleda (Pontevedra)

Tfn. 986581413; fax 986580162 <secretgati@ati.es>

Suscripción y Ventas

<<http://www.ati.es/novatica/interes.html>>, o en ATI Cataluña o ATI Madrid

Publicidad

Padilla 66, 3º, dcha., 28006 Madrid

Tfn. 914029391; fax 913093685 <novatica.publicidad@ati.es>

Imprenta

Derra S.A., Juan de Austria 66, 08005 Barcelona

Dedición legal: B 13.134-1975 -- ISSN: 0211-2124; CODEN NOVACE

Portada: Antonio Crespo Foix / © ATI 2005

Diseño: Fernando Agrieta / © ATI 2005

editorial	> 02
Las patentes de Software: el gran revolcón	
Andalucía: otra Ley de Colegios excluyente e inoperante	
en resumen	> 02
El Software Libre en el diván	
<i>Rafael Fernández Calvo</i>	
monografía	
El Software Libre como objeto de estudio	
(En colaboración con UPGRADE y con la cooperación del proyecto europeo CALIBRE)	
Editores invitados: <i>Jesús M. González Barahona, Stefan Koch</i>	
Presentación. El Software Libre al microscopio	> 03
<i>Jesús M. González Barahona, Stefan Koch</i>	
CALIBRE en la cresta de la ola europea del Software de Código Abierto	> 05
<i>Andrea Deverell, Par Agerfalk</i>	
¿Será el movimiento del Software Libre el nuevo escalón en el modelo de organización de la producción en el sector TI?	> 06
<i>Nicolas Jullien</i>	
Debian 3.1 (Sarge) como caso de estudio de medición del Software Libre: resultados preliminares	> 11
<i>Juan José Amor Iglesias, Jesús M. González Barahona, Gregorio Robles Martínez, Israel Herráiz Taberero</i>	
El análisis institucional aplicado al estudio del Software Libre como "bien comunal"	> 15
<i>Charles M. Schweik</i>	
Sobre proyectos de Software Libre / Código Abierto de "puerta cerrada": enseñanzas del enfoque de selección de desarrolladores para Firefox de Mozilla	> 23
<i>Sandeep Krishnamurthy</i>	
Agilidad y desarrollo de Software Libre	> 27
<i>Alberto Sillitti, Giancarlo Succi</i>	
secciones técnicas	
Enseñanza Universitaria de la Informática	
Propuesta de objetivos formativos para el primer curso de las Ingenierías Informáticas y de algunas estrategias docentes para conseguirlos	> 31
<i>Fermín Sánchez Carracedo, Ricard Gavalà Mestre</i>	
Gestión del Conocimiento	
Escenarios del conocimiento: conocimiento orgánico e inorgánico	> 36
<i>Juan Baiget Solé</i>	
Ingeniería del Software	
La gestión de la diversidad de procesos por los informáticos: reflexiones	> 38
<i>Daniilo Caivano, Corrado Aaron Visaggio</i>	
Aspectos pragmáticos en el Desarrollo por el Usuario Final	> 45
<i>José Antonio Macías Iglesias</i>	
Lenguajes informáticos	
Mono: mucho más que una implementación libre de .Net	> 48
<i>Jordi Mas i Hernández</i>	
Lingüística computacional	
Procesamiento y aplicaciones de los corpus paralelos	> 50
<i>Xavier Gómez Guinovart</i>	
Redes y servicios telemáticos	
Extensión del servicio A/V Streaming de CORBA. Modelo empírico basado en el control de tráfico	> 55
<i>Antonio Javier García Sánchez, Felipe García Sánchez, Pablo Pavón Mariño, Joan García Haro</i>	
Encaminamiento inter-dominio con calidad de servicio basado en Overlay Entities distribuidas y QBGP	> 61
<i>Marcelo Yannuzzi, Alexandre Fonte, Xavier Masip Bruin, Edmundo Monteiro, Sergi Sánchez López, Marilía Curado, Jordi Domingo Pascual</i>	
Referencias autorizadas	> 68
sociedad de la información	
Programar es crear	
Un evento que mejora cada año: el Concurso Universitario de Programación de la Comunidad Autónoma de Madrid (CUPCAM)	> 74
<i>Adolfo Vázquez Rodríguez</i>	
Dominó Solitario (CUPCAM 2005, problema A)	> 75
<i>Antonio Fernández Anta</i>	
asuntos interiores	
Coordinación editorial / Programación de Novática	> 76
Normas de publicación para autores / Socios Institucionales	> 77

Sandeep Krishnamurthy
 Universidad de Washington, Bothell (EE.UU.)

<sandeep@u.washington.edu>

Envié al club un telegrama que decía: "ACEPTE POR FAVOR MI DIMISIÓN. NO DESEO PERTENECER A NINGÚN CLUB QUE ME ACEPTA COMO MIEMBRO."

Groucho Marx.
 cómico estadounidense (1890 - 1977)

1. Introducción

La gran mayoría de proyectos de código abierto son 'pequeños', es decir, tienen menos de cinco miembros [3][5][4][11]. Muchos proyectos de código abierto son 'cuevas' o cuentan con un único miembro [7].

Al mismo tiempo, algunos estudiosos del código abierto han sostenido que el número de desarrolladores da una idea del nivel de éxito de ese proyecto [9]. En esta línea de razonamiento, puesto que los desarrolladores tienen opciones múltiples y muchas demandas de su tiempo, la mera atracción de una gran cantidad de desarrolladores por un proyecto es una indicación de éxito. En [12] se ha argumentado que los proyectos que no crecen más allá de un "núcleo fuerte de desarrolladores ... fallarán debido a una carencia de los recursos dedicados a encontrar y a corregir errores" (pp. 329-341). Sin embargo este argumento de "cuanto mayor mejor" ignora algunos de los aspectos negativos de atraer un número alto de desarrolladores para un proyecto; por ejemplo, los costes de coordinación, el conflicto, la pérdida de calidad.

Lo que ahora estamos aprendiendo es que el mandato recogido en "La Catedral y el Bazar" [14] --"tratar a tus usuarios como co-desarrolladores es el camino más sencillo para mejorar el código de forma rápida y corregir errores con efectividad"-- no siempre es aplicable. Se observa que algunos grupos no van en la línea del modelo promiscuo 'raymondiano' de "usuario como co-desarrollador", que permite el libre acceso al código y amplios privilegios de revisión. Más bien vemos que no se solicita la opinión de los individuos y un grupo cerrado controla las tareas más importantes (por ejemplo, la revisión de código).

En este artículo, usando el ejemplo del navegador Firefox de Mozilla, argumento que algunos proyectos muy exitosos de Software Libre / Código Abierto (FLOSS, Free/Libre/Open Source Software) están diseñados para ser pequeños. Lejos de buscar una gran cantidad de desarrolladores, estos grupos desalientan activamente a los aspirantes e incluso no permiten que los individuos

Sobre proyectos de Software Libre / Código Abierto de "puerta cerrada": enseñanzas del enfoque de selección de desarrolladores para Firefox de Mozilla

© El presente artículo está protegido por derechos de autor según la licencia *Creative Commons Attribution-NonCommercial-NoDerivs 2.5*, que se puede consultar en <<http://creativecommons.org/licenses/by-nc-nd/2.5/>>

Traducción: José Ventura Roda (Socio de ATI), José Zurita Márquez (Socio de ATI, miembro del Grupo de Lengua e Informática de ATI)

Resumen: en este artículo introduzco la noción de un proyecto de código abierto de puerta cerrada. En tales proyectos, las tareas más importantes del desarrollo (por ejemplo, revisión del código) son controladas por un grupo restringido. Aquí presento cinco nuevos argumentos sobre por qué hay grupos que pueden desear organizarse de esta manera. La primera razón es que los desarrolladores simplemente no tienen tiempo disponible para evaluar a los aspirantes. Los dos argumentos siguientes están basados en la autoselección, pues mediante el establecimiento de requisitos exigentes de entrada, el proyecto puede asegurarse de que se consiguen programadores de calidad y con mucha constancia. El cuarto argumento es que la ampliación de un grupo destruye la diversión. La quinta razón es que los proyectos que requieren conocimientos variados precisan un enfoque de puerta cerrada.

Palabras claves: Firefox, puerta cerrada, software de código abierto, Software Libre, selección de desarrolladores, tamaño del grupo, cueva.

Autor

Sandeep Krishnamurthy es Profesor Asociado de comercio electrónico y marketing en la Universidad de Washington, Bothell (EE.UU.). En la actualidad centra sus estudios en el impacto de Internet sobre las empresas, las comunidades y las personas. Es autor de un libro de texto muy exitoso sobre comercio electrónico para cursos de administración de empresas titulado "E-Commerce Management: Text and Cases" y ha publicado recientemente la obra "Contemporary Research in E-Marketing: Volumes I, II". Sus trabajos académicos han sido publicados en revistas como *Organizational Behavior and Human Decision Processes (OBHDP)*, *Marketing Letters*, *Journal of Consumer Affairs*, *Journal of Computer-Mediated Communication*, *Quarterly Journal of E-Commerce*, *Marketing Management*, *Information Research*, *Knowledge, Technology & Policy* y *Business Horizons*. Es editor asociado revisor de libros de *Journal of Marketing Research* y ha sido coeditor de una monografía de *International Marketing Review* sobre e-Marketing. Algunos de sus artículos han aparecido en sitios web especializados como *Clickz.com*, *Digitrends.net* y *Marketingprofs.com*. Ha intervenido recientemente en importantes emisoras de radio y televisión para señalar los errores del programa de corrección de errores gramaticales de Microsoft. Sus comentarios han ido reproducidos por diversos medios de comunicación. Trabaja asimismo en las áreas de marketing genérico y marketing no lucrativo. Su sitio web está en <<http://faculty.washington.edu/sandeep>> y su cuaderno de bitácora en <<http://sandeepworld.blogspot.com>>.

interesados envíen parches. En lugar de abrir las puertas a todos los interesados, estos proyectos proporcionan el código de sus programas al mundo entero, pero no permiten que cualquiera participe en el desarrollo del producto. Basándome en conversaciones *on-line* públicas, proporciono cinco explicaciones teóricas para explicar por qué algunos grupos de fuente abierta toman la vía de "puerta cerrada".

2. El equipo de desarrollo de Firefox

El navegador Firefox, de la Fundación Mozilla, <<http://www.mozilla.org/products/firefox/>>, funciona muy bien. Ha sido descargado más de 60 millones de veces en un período muy corto de tiempo. Aunque el navegador Firefox se beneficia del extenso

código de Netscape, fue un equipo muy pequeño de programadores comprometidos el que desarrolló el navegador Firefox completo. En este momento, el grupo base lo forman seis personas: Blake Ross, David Hyatt, Ben Goodger, Brian Ryner, Vladmir Vukicevic y Mike Connor. En todos los sentidos, un grupo pequeño ha tenido el privilegio de la revisión del código. El proyecto Firefox (originalmente llamado Phoenix y después retitulado Firebird) lo inició Blake, que durante mucho tiempo había solucionado fallos en el navegador de Mozilla y estaba desencantado con la dirección del proyecto. David Hyatt entró en el proyecto, puesto que era ex-empleado de Netscape y tenía un conocimiento profundo del código. Un subconjunto significativo de este grupo base

recibía un sueldo de la Fundación Mozilla por trabajar en este proyecto.

Los documentos redactados por los miembros de este equipo nos permiten observar de forma poco habitual los motivos que llevan a algunos grupos a mantener a otros a distancia. Véanse estos extractos de las preguntas más frecuentes (FAQ) en el manifiesto original del equipo (la fuente es <<http://www.blakeross.com/firefox/README-1.25.html>>):

■ Q2. ¿Por qué solamente un equipo pequeño?

■ El tamaño del equipo que trabaja en el núcleo es una de las muchas razones por las que el desarrollo del núcleo es tan lento. Tenemos la sensación de que menores dependencias (sin restricciones de marketing), una innovación más rápida (ningún comité de interfaz de usuario) y más libertad para experimentar (sin requisitos de compatibilidad con versiones anteriores) conducirán a un producto final mejor.

■ Q3. ¿Dónde informo sobre los errores de este paquete?

■ Estamos todavía trabajando con trazo grueso. Hay muchas partes que obviamente están mal y no necesitamos que nos informen sobre errores en las mismas. Si encuentra un error (solicitar una prestación no es un error) y está seguro de que es específico de Firefox (no aparecía en Mozilla) y ha leído suficientemente a fondo todos los errores existentes en Firefox para saber que no está ya registrado, siéntase libre para comunicarlo en el producto Phoenix en Bugzilla. (...)

■ Q5: ¿Cómo puedo incorporarme?

■ Por invitación. Esto es una meritocracia: se invitará a sumarse al grupo a los que se ganen el respeto del grupo.

Las respuestas son muy claras. Desalientan a los que desean participar en el proceso. Los participantes potenciales son advertidos de que se es miembro solo por invitación. Claramente, el grupo ha hecho todo lo posible para dejar fuera a la gente en lugar de animarla a participar. Si la captación de desarrolladores es el camino para el éxito de un proyecto de código fuente abierto [9], éste no sería el caso.

Esto nos enseña que algunos proyectos de código abierto no son proyectos de "puerta abierta". Muchos se pueden describir mejor como proyectos de puerta cerrada. Formalmente, se definen los proyectos de puerta cerrada como los que proporcionan acceso al programa y al código fuente a cualquier persona interesada, pero no facilitan el acceso a las funciones nucleares del software desarrollado (especialmente a la determinación de las líneas estratégicas, la comprobación del código y el envío de parches).

Tales proyectos desean que su público objetivo

lo descargue y utilice su producto y trate de corregir su código. Sin embargo, mantienen a distancia a potenciales participantes cualificados. Incluso no admiten en el equipo a los desarrolladores que trabajan en errores y envían parches. Firefox se ha desarrollado coherentemente como un proyecto de puerta cerrada. Si bien los usuarios pueden ofrecer sus opiniones en foros abiertos, el desarrollo real está hecho por un grupo base.

Este método es polémico y trastorna a muchos miembros de la comunidad de código abierto. Si los proyectos de código abierto se basan en formar una comunidad de expertos, el modelo de puerta cerrada proporciona aparentemente una manifestación de esta filosofía construida sin embargo sobre los principios de control habituales. He aquí una reacción pública a la política de reclutamiento para el desarrollo de Firefox.

Dicen en voz alta que solamente están dispuestos a aceptar en el proyecto a desarrolladores aprobados por ellos, no necesitan a nadie que se presente. Y con esta actitud, alejan a gente que quiere ayudar pero que no está segura de sus aptitudes.

Después dicen que quieren que la gente envíe parches y que echen una mano en el desarrollo del proyecto. ¿Pero cómo suponen que alguien va a hacer esto sin ser miembro? Bien, obviamente usted no tiene que estar en el equipo para trabajar por el equipo. ¿Pero quién desea trabajar para alguien que no va a tratarlo como parte del mismo equipo?

...
Sin embargo, el espíritu del código abierto (al menos de la parte BSD) es de apertura y aceptación. No permitir la entrada de la gente o aceptar a los nuevos miembros solo a través de invitación huele a elitismo. Desafortunadamente cuando te relacionas con otros seres humanos, inevitablemente terminarás encontrando alguno que cree pertenecer a la élite y merecedor de mirar al resto por encima del hombro. (Fuente: <<http://developers.slashdot.org/comments.pl?sid=137815&cid=11526872>>).

Por otra parte, confiar en un pequeño grupo podría potencialmente poner en peligro el futuro de un proyecto si sus componentes tienen otras oportunidades. Mike Connor, uno de los principales desarrolladores de Firefox expresó su frustración de esta manera: *"Esto me está molestando, y me ha estado molestando desde hace tiempo. Por muchas razones, en los tres años que llevamos trabajando con el proyecto, no hemos logrado construir una comunidad de desarrolladores expertos en torno a Firefox (el subrayado es mío). Creo que ahora tenemos problemas. De las seis personas que controlan realmente Firefox, cuatro están ausentes sin permiso y uno no hace demasiadas revisiones. Y yo estoy a punto de irme indefini-*

damente, puesto que siento que soy la única persona a la que todo esto que le preocupa lo suficiente para convertirlo en un problema". (Fuente: <<http://steelgrypnon.com/blog/index.php?p=37>>)

3. La teoría de la cebolla

En estos momentos, la teoría de la cebolla sobre el desarrollo de software de código abierto se ha puesto de moda (por ejemplo, [13][15][3][10]). En esta teoría, un grupo pequeño de individuos con poder controlan los privilegios de revisión, mientras que asignan a los miembros de las capas externas de la cebolla tareas rutinarias como la corrección de errores. Las razones y las implicaciones de esta estructura de organización todavía no se han explicado por completo en la literatura especializada.

La explicación más común para restringir el tamaño del grupo es que incrementar el número de desarrolladores plantea problemas de coordinación [2]. Algunos estudiosos han precisado que el acceso al grupo base en la mayoría de los proyectos de código abierto es controlado por un proceso automático de alta. Los que saben cómo contactar a los desarrolladores del proyecto de una manera culturalmente compatible consiguen entrar, mientras que a otros se les niega el acceso [15]. Los procesos de alta automática desempeñaron ciertamente su papel en la elección de los desarrolladores del grupo base de Firefox --Dave Hyatt fue empleado de Netscape y conocía desde luego su cultura, Ben Goodger se incorporó como resultado de una cuidadosa crítica del navegador de Mozilla en su sitio web.

En este artículo, construido a partir de conversaciones públicas sobre Firefox, discuto otras cinco explicaciones para implementar un método de puerta cerrada. No queda claro, por supuesto, si todas las explicaciones proporcionadas aquí se aplican a todos los proyectos de puerta cerrada. Una investigación futura debería evaluar la importancia relativa de cada explicación. Estos argumentos son, de alguna manera, nuevos en la manera que se presentan y deben mover al diálogo sobre el reclutamiento de los desarrolladores de código abierto en lo sucesivo.

4. Cinco explicaciones

Explicación 1: poco tiempo disponible

Evaluar a nuevos miembros lleva un tiempo del que los desarrolladores no disponen. Así resulta, con frecuencia, que los líderes, que tienen un tiempo limitado a su disposición, eligen hacer el trabajo ellos mismos mejor que gastar el tiempo intentando identificar a un potencial candidato. Blake Ross ha dicho que *"Ahora mismo solo busco tiempo para estar trabajando en Firefox, a pesar de esto, a menudo tengo 1-2 horas máximo (al día). Ben, obviamente, tiene sus manos ocupadas*



Los proyectos de "puerta cerrada" son iniciados con frecuencia por un pequeño grupo de personas con relaciones de confianza entre ellos



liderando e intentando que todos los patos estén en fila ". (Fuente: <<http://blakeross.com/index.php?p=19>>)

Encontrar a nuevos miembros del equipo es una tarea onerosa y arriesgada. La tarea implica el coste de anunciarse y de examinar a los aspirantes. Si el aspirante no conoce a un miembro del grupo base, puede ser difícil juzgar la competencia y las capacidades de nuevos miembros potenciales. Por otra parte, se presenta generalmente el llamado "problema de la agencia", según el cual los aspirantes pueden ocultar sus habilidades o jugar esta baza como una vía para asegurar su incorporación. Por lo tanto, cuando a los miembros les falta tiempo, el beneficio de incluir un nuevo miembro se contrarresta con el coste de buscar a esa persona.

Explicación 2: meritocracia (sólo el más experto consigue entrar)

Las comunidades de código abierto compiten con las empresas por los desarrolladores. El atraer a los mejores desarrolladores aumenta las posibilidades de la comunidad de competir en un duro mercado. Establecer los niveles más altos les permite reclutar a desarrolladores altamente cualificados, mejorando sus posibilidades de éxito. El cierre de puertas lleva a la autoselección, contando con los desarrolladores interesados más competentes. Es posible determinar la calidad del trabajo de un desarrollador de código abierto, puesto que el resultado de una tarea es libre y por eso, fácilmente accesible.

Esta teoría es apoyada por una observación hecha por Blake Ross, un desarrollador clave de Firefox: "Básicamente, utilizamos el código abierto como la mejor entrevista de trabajo del mundo. En vez de entrevistar a la gente delante de una mesa de despacho durante dos horas y pedirle que muevan el monte Fuji (el autor aclara que es una cita de un libro sobre los procedimientos de entrevistas de Microsoft), queremos que la gente envíe código que pueda demostrar exactamente lo que podría aportar si se unieran al equipo ". (Fuente: <<http://blakeross.com/index.php?p=19>>)

Un participante en una discusión en Slashdot también formuló un argumento similar: "En Firefox buscan a los 'programadores más inteligentes' para trabajar con su código base. Aunque está claro que es elitista, se aseguran de que sólo la élite (dedicación

más habilidad) consigue trabajar en los entresijos del navegador. Si esto termina haciéndoles trabajar más rápidamente, con más fiabilidad y más eficiencia, entonces mejor que mejor. Un pequeño equipo de individuos con mucha experiencia suele lograr más que un gran grupo de gente con una experiencia media y habitualmente llegarán más lejos que un equipo grande de personas con un perfil mediocre. Se puede garantizar que todos los que compiten con ellos (empresas como MS y Opera) son bastante elitistas (contratarán mediante entrevistas a los mejores codificadores y diseñadores que puedan), así que, ¿por qué no debería hacerlo el equipo de Firefox? Por supuesto, como se ha indicado, si piensas que lo puedes hacer mejor con tu forma de reclutar un equipo, entonces crea tu propio proyecto, y veremos cual sobrevive ". (Fuente: <<http://developers.slashdot.org/comments.pl?sid=137815&cid=11527401>>)

Explicación 3: persistencia (sólo los más persistentes logran entrar)

Los proyectos de puerta cerrada disuaden a la gente de aspirar a entrar. Frecuentemente se hace referencia a la cantidad de trabajo que conlleva. Véase, por ejemplo, este mensaje de Ben Goodger de Firefox de Mozilla: "Se necesita ayuda. Necesitamos 'grandes productores' de código. Si estás interesado en la tecnología de los navegadores web, ¿por qué involucrarte en el proyecto de navegador de código abierto más importante? Buscamos sobre todo personas con experiencia de programación en Mac OS X y desarrolladores en Windows. Comienza hoy, buscando y arreglando algo. Aquí no se facilita ninguna formación, puesto que averiguar cómo se hace todo esto puede considerarse parte de los 'requisitos de entrada';-)". (Fuente: <<http://www.mozilla.org/projects/firefox/>>)

Debe ser el anuncio de petición de ayuda más intimidatorio del mundo. Ben, literalmente, dice a la gente que deben trabajar duro a cambio de nada y si quieren impresionarle, deberían empezar a trabajar por abajo, por ejemplo, solucionando errores.

Blake Ross justifica este enfoque de esta manera: "Ben admite que percatarse de cómo hacerse notar es parte del proceso de reclutamiento, y realmente lo es. Después de todo, muchos de los actuales super revisores de Mozilla y las personas que llevan el proyecto empezaron como colaboradores 'no-

vatos' y escalaron hasta la cima de la meritocracia. Si no estás dispuesto a realizar un poco de investigación y a imaginar como dejar tu huella, ¿de verdad estás en el equipo? ". (Fuente: <<http://blakeross.com/index.php?p=19>>)

Esto llevará probablemente a las personas muy persistentes a autoseleccionarse en el proyecto con resultados positivos. Mientras que la autoselección basada en la calidad del desarrollador es comprensible, la autoselección basada en la persistencia puede conducir a que sean admitidos programadores de baja calidad (por ejemplo, aquéllos que tienen mucho tiempo disponible). Aunque una dedicación prolongada al proyecto (por ejemplo, con muchos mensajes en los foros) es una señal de persistencia, puede ser también un indicador de fervor ideológico. Por lo tanto, permitir la autoselección a través de la persistencia puede llevar a peculiaridades en términos de composición del grupo. Otros grupos han solucionado la cuestión de los participantes persistentes que no aportan mucho mediante la creación de listas electrónicas específicas de desarrolladores [12].

Explicación 4: abriendo la puerta se puede matar la diversión

Los eruditos que estudian la motivación de los desarrolladores de código abierto nos dicen que muchos individuos están motivados por la diversión que aporta construir algo [8][6]. En [1] incluso se califica a los desarrolladores de código abierto como *homo ludens*, y modelo de motivación intrínseca de desarrolladores. Eric Raymond, un veterano observador del FLOSS y autor del popular "The Cathedral and the Bazaar", ha dicho que: "puede resultar que uno de los más importantes efectos del éxito del código abierto sea enseñarnos que el juego es el modo económicamente más eficiente de realizar trabajo creativo" ¹.

Los proyectos de puerta cerrada son iniciados con frecuencia por un pequeño grupo de personas con relaciones de confianza entre ellos. Admitir a extraños elimina esta sensación acogedora y reduce la motivación intrínseca de los desarrolladores actuales.

Blake Ross ha comentado que Firefox siempre ha sido un proyecto familiar: "A veces la gente pregunta por qué trabajamos gratis en Firefox. Es difícil contener la risa cuando dicen 'trabajo'. Dime otro proyecto que to-

que la vida de millones de personas en todo el mundo y que tenga un nombre clave como 'The Ocho', que consiga ser publicado en todos los medios". ('The Ocho' es el nombre familiar del canal de televisión ficticio ESPN8 en la película Dodgeball; puntos para Ben por su inspiración al inventar un nombre brillante para la versión 1.5).

Lo mejor de Firefox es que aunque ha subido como un cohete, nunca ha olvidado sus humildes raíces como proyecto 'canalla' que era, coordinado con altas dosis de cafeína en Denny's. En resumen, el proyecto nunca será totalmente adulto. (Fuente: <<http://blakeross.com/index.php?p=24>>)

Como un observador dijo en Slashdot: "Quizás es cuestión de divertirse...". Si tu limitas los desarrolladores a personas que realmente disfrutan trabajando juntas y tienen ideas similares sobre cómo comportarse y hablar con la gente, puede ser más divertido que si invitas a todos los codificadores inadaptados socialmente, que no pueden tomarse el rechazo de un parche más que como un insulto (o, para los verdaderamente chiflados, como un cierto juego político).

Hay más de un par de grandes codificadores fuera de aquí con habilidad cero para tratar con la gente. Pueden dañar un proyecto porque, aunque sus propias contribuciones son buenas, bajan el nivel de diversión y por lo tanto la productividad de todos.

Algunos de ellos hacen grandes proyectos en plan solista ... (Fuente: <<http://developers.slashdot.org/comments.pl?sid=137815&cid=11527896>>)

Observaciones similares se han oído en el ámbito de los emprendedores. Con frecuencia encontramos que una compañía la pone en marcha un grupo de buenos amigos. Con el tiempo, según aumenta la formalización, la compañía introduce más procedimientos, haciéndolos más estructurados y burocráticos, perdiendo su naturaleza *ad hoc* y divertida. No está claro si esto puede ser solamente una explicación parcial de la existencia de los proyectos de puerta cerrada.

Explicación 5: los productos que requieren capacidades variadas requieren el enfoque de puerta cerrada

Firefox es uno de los pocos productos de código abierto dirigido a una audiencia general; es decir, a un mercado de consumo. A diferencia de la inmensa mayoría de los productos de código abierto dirigidos a una audiencia general, Firefox necesita tener éxito con los consumidores no profesionales. Esto implica que para que el proyecto tenga éxito necesita un interfaz de usuario intuitivo junto con un producto robusto.

Por lo tanto, el equipo de proyecto necesita

implicar a gente con capacidades diversas, algunos que sean más expertos en el diseño de interfaces de usuario y otros que tengan grandes conocimientos de programación.

El manifiesto original indica que: "tenemos la sensación de que menores dependencias (sin restricciones de marketing), una innovación más rápida (ningún comité de interfaz de usuario) y más libertad para experimentar (sin requisitos de compatibilidad con versiones anteriores) conducirán a un producto final mejor".

En palabras de Blake Ross: "Puesto que esta audiencia era principalmente de carácter no técnico, estimamos necesario para juzgar las mejoras no sólo su mérito técnico, sino también cómo se alineaban con esta nueva visión. La revisión del código más la interfaz de usuario llevó, sin embargo, mas tiempo del que estábamos dispuestos a gastar en nuestra impaciencia por desarrollar rápidamente Phoenix. Así que intentamos encontrar a las personas que comprendieran nuestra visión tan bien que no necesitaran capas adicionales de revisión y las incorporamos al equipo". (Fuente: <<http://blakeross.com/index.php?p=19>>)

Por lo tanto, la naturaleza única de Firefox hizo necesario un enfoque de puerta cerrada. El grupo no quería un comité de interfaz de usuario y quiso manejar los parches de forma diferente (es decir, revisión de código más interfaz de usuario). El aumento del tamaño del grupo y la incorporación de extraños hubieran diluido este proceso.

5. Conclusión

En este artículo he propuesto cinco nuevos argumentos para organizar un proyecto de código abierto como puerta cerrada. El primer argumento es que los desarrolladores simplemente no tienen tiempo disponible para evaluar a los miembros potenciales y prefieren utilizar su tiempo en sacar trabajo, mejor que en invertirlo en evaluar nuevos miembros. Los siguientes dos argumentos están basados en la autoselección, pues mediante la definición de requisitos difíciles para entrar el proyecto se puede garantizar que se consiguen programadores de gran calidad y muy persistentes. El cuarto argumento aplica la motivación intrínseca inducida por la diversión (*homo ludens*) y que implica que extender un grupo más allá de un pequeño clan arruina la diversión. El quinto argumento es que los proyectos complicados (por ejemplo, aquellos que requieren conocimientos en áreas técnicas y de interfaz de usuario) precisan un enfoque de puerta cerrada.

La investigación futura debe estudiar la importancia relativa de estos argumentos. Por otra parte, no sabemos si los resultados del proyecto son mejores o peores por organizarse al modo puerta cerrada. Es necesaria pues una comparación empírica de los enfoques de puerta cerrada y puerta abierta.

Referencias

- [1] J. Bitzer, W. Schrettl, P.J.H. Schroeder. "Intrinsic Motivation in Open Source Software Development", 2004. Disponible en <<http://econwpa.wustl.edu/eps/dev/papers/0505/0505007.pdf>>.
- [2] Frederick Brooks. *The Mythical Man-Month: Essays on Software Engineering, 20th Anniversary Edition*, Addison-Wesley Professional, 1995.
- [3] K. Crowston, J. Howison. "The Social Structure of Open Source Software Development Teams. Working Paper", 2003. Disponible en <<http://floss.syr.edu/tiki-index.php>> [accedido el 22-8-2004].
- [4] K. Healy, A. Schussman. "The Ecology of Open Source Software Development", 2004. Documento de trabajo no publicado. Disponible en <<http://www.kieranhealy.org/files/drafts/ossactivity.pdf>> [accedido el 22-8-2004].
- [5] F. Hunt, P. Johnson. "On the Pareto distribution of Sourceforge projects", en *Proceedings of the F/OSS Software Development Workshop*. 122-129, Newcastle, UK, 2002.
- [6] S. Krishnamurthy. "On the Intrinsic and Extrinsic Motivation of Open Source Developers", 2005. De próxima aparición en *Knowledge, Technology & Policy*.
- [7] S. Krishnamurthy. "Cave or Community?: An Empirical Examination of 100 Mature Open Source Projects", en *First Monday*, 7(6), 2002. Disponible en <http://firstmonday.org/issues/issue7_6/krishnamurthy/index.html>.
- [8] K.R. Lakhani, R. Wolf. "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects", en *Perspectives on Free and Open Source Software*, editado por J. Feller, B. Fitzgerald, S. Hissam y K. R. Lakhani. Cambridge, MA: MIT Press, 2005.
- [9] J. Lerner, J. Tirole. "Some Simple Economics of Open Source" en *Journal of Industrial Economics*, 52, pp. 197-234, 2002.
- [10] Luis Lopez-Fernandez, Luis, Gregorio Robles, Jesus M. González-Barahona. "Applying Social Network Analysis to Information in CVS Repositories", 2005. Disponible en <<http://opensource.mit.edu/papers/llopez-sna-short.pdf>>.
- [11] G. Madey, V. Freeh, R. Tynan. "Modeling the F/OSS Community: A Quantitative Investigation", en *Free/Open Source Software Development*, editado por Stephan Koch, Hershey, PA: Idea Group Publishing, 2004.
- [12] A. Mockus, R. T. Fielding, J. D. Herbsleb. "Two Cases of Open Source Software Development: Apache and Mozilla", en *ACM Transactions on Software Engineering and Methodology*, 11(3), pp. 309-346, 2002.
- [13] K. Nakakoji, Y. Yamamoto, Y. Nishinaka, K. Kishida, Y. Ye. "Evolution Patterns of Open-Source Software Systems and Communities", en *Proceedings of International Workshop on Principles of Software Evolution (IWPE 2002)*, pp. 76-85, 2002.
- [14] E. Raymond. "The Cathedral and the Bazaar", en *First Monday*, volumen 3, número 3, 1998. Disponible en <http://www.firstmonday.org/issues/issue3_3/ramond/>.
- [15] G. Von Krogh, S. Haefliger, S. Spaeth. "Collective Action and Communal Resources in Open Source Software Development: The Case of Freenet", en *Research Policy*, 32(7), pp. 1217-1241, 2003.

Nota

¹ Agradecemos a Bitzer, Schrettl y Schroeder (2004), por avisarnos de la cita de la página 9 de la obra de Raymond.